

CLAIMS

1. A method for improving execution performance of a repeated
sequence of instructions that provide a function and having external
access points that are external entry and external exit points, comprising
the steps of:

determining at least one instruction, from the sequence of
instructions, that is necessary to be executed for less than all repetitions
of the sequence of instructions; and

modifying the sequence of instructions to isolate the one instruction
from only some of the repetitions of the sequence of instructions.

2. The method of claim 1, wherein:

said modifying includes the step of inserting at least one internal
access point within the sequence of instructions and thereby partitioning
the sequence of instructions into multiple segments, and having one of the

multiple segments including the one instruction and executing for fewer times than the number of executions of another of the multiple segments.

3. The method of claim 2, wherein

said inserting step inserts the one internal access point as an

internal recursive entry point.

4. The method of claim 3, wherein

said modifying includes the step of moving the one instruction from

outside of the one of the multiple segments to within the one of the

multiple segments and between one of the external access points and the

internal recursive access point.

5. The method of claim 2, wherein

said modifying includes the step of moving the one instruction from

outside of the one of the multiple segments to within the one of the multiple segments and between one of the external access points and the internal access point.

6. The method of claim 1, wherein

said modifying includes the step of rescheduling the one instruction closer in sequence of execution to one of the external access points.

7. A computer readable storage media having computer readable code physically implementing a method of improving execution performance of a sequence of instructions, the code including statements for performing the method of claim 1.

8. A computer readable storage media having computer readable code physically implementing a method of improving execution

performance of a sequence of instructions, the code including statements
for performing the method of claim 2.

9. A computer readable storage media having computer readable
code physically implementing a method of improving execution
performance of a recursive sequence of instructions, the code including
statements for performing the method of claim 5.

10. A computer readable storage media having computer readable
code physically implementing a method of improving execution
performance of a sequence of instructions, the code including statements
for performing the method of claim 6.

11. A computer system including the computer readable storage
media of claim 7, further comprising:

at least one processing unit coupled to said computer readable storage media for executing the sequence of instructions of the computer readable code; and

said computer readable storage media including at least one of volatile and non-volatile memory.

12. A computer system including the computer readable storage media of claim 8, further comprising:

at least one processing unit coupled to said computer readable storage media for executing the sequence of instructions of the computer readable code; and

said computer readable storage media including at least one of volatile and non-volatile memory.

13. A computer system including the computer readable storage media of claim 9, further comprising:

HAL 191

to be executed for less than all repetitions of the program, comprising:

executing at least some of the sequence of instructions from an externally called entry point in the program initially;

thereafter repeatedly calling the program;

5 in response to said repeatedly calling, executing only some of the sequence of instructions;

thereafter exiting the program from an exit point; and

controlling at least one of said steps of executing with an internal access point other than the entry point and the exit point to isolate the one instruction within the sequence of instructions from at least one of said repeatedly calling and to execute the one instruction a number of times fewer than the total number of executions of the entire sequence of instructions.

15 16. A method of machine executing according to claim 15, wherein:

said first-mentioned executing, includes executing the one

instruction;

said internal access point is an internal recursive entry point

scheduled after the one instruction in the sequence of instructions; and

said second-mentioned executing recursively starts from the internal recursive entry point.

5

17. A method of processing, comprising:

providing a sequence of instructions repeatable to perform a function and having at least one instruction that is necessary to be executed for less than all repetitions of the sequence of instructions; and

providing an internal access point other than an externally called entry point and an external exit point, which internal access point isolates the one instruction within the sequence of instructions from only some of the repetitions so that the one instruction is within less than all of the repetitions.

15

HAL 191

18. The method of claim 17, wherein all of said steps are included within a step of storing a program.

19. The method of claim 17, wherein all of said steps are included within a step of transmitting a program.

20. The method of claim 17, wherein all of said steps are included within a step of receiving a program.

21. The method of claim 17, wherein all of said steps are included within a step of executing a program.

22. The method of claim 17, wherein all of said steps are included within a step of machine modifying a program.

5

10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100
105
110
115
120
125
130
135
140
145
150
155
160
165
170
175
180
185
190
195
200
205
210
215
220
225
230
235
240
245
250
255
260
265
270
275
280
285
290
295
300
305
310
315
320
325
330
335
340
345
350
355
360
365
370
375
380
385
390
395
400
405
410
415
420
425
430
435
440
445
450
455
460
465
470
475
480
485
490
495
500

15

means for rescheduling at least one instruction of a repeated

5

said means for rescheduling providing internal recursive access

	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100	2101	2102	2103	2104	2105	2106	2107	2108	2109	2110	2111	2112	2113	2114	2115	2116	2117	2118	2119	2120	2121	2122	2123	2124	2125	2126	2127	2128	2129	2130	2131	2132	2133	2134	2135	2136	2137	2138	2139	2140	2141	2142	2143	2144	2145	2146	2147	2148	2149	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159	2160	2161	2162	2163	2164	2165	2166	2167	2168	2169	2170	2171	2172	2173	2174	2175	2176	2177	2178	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191	2192	2193	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207	2208	2209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223	2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253	2254	2255	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269	2270	2271	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284	2285	2286	2287	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298	2299	2300	2301	2302	2303	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313	2314	2315	2316	2317	2318	2319	2320	2321	2322	2323	2324	2325	2326	2327	2328	2329	2330	2331	2332	2333	2334	2335	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	2346	2347	2348	2349	2350	2351	2352	2353	2354	2355	2356	2357	2358	2359	2360	2361	2362	2363	2364	2365	2366	2367	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377	2378	2379	2380	2381	2382	2383	2384	2385	2386	2387	2388	2389	2390	2391	2392	2393	2394	2395	2396	2397	2398	2399	2400	2401	2402	2403	2404	2405	2406	2407	2408	2409	2410	2411	2412	2413	2414	2415	2416	2417	2418	2419	2420	2421	2422	2423	2424	2425	2426	2427	2428	2429	2430	2431	2432	2
--	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	---